

CLAIMS

What is claimed is:

1. An arbitration unit, comprising:
 - an allocation unit that assigns a first number of slots in a frame to a first flow and a second number of slots in the frame to a second flow; and
 - a scheduling unit that assigns first slot positions to the first number of slots in the frame and second slot positions to the second number of slots in the frame using a binary distribution tree.
2. The arbitration unit of Claim 1, wherein the first flow and the second flow are flows of discrete packets.
3. The arbitration unit of Claim 1, wherein the binary distribution tree generates a binary permission vector (BPV) for each of the first and second flows.
4. The arbitration unit of Claim 1, wherein the scheduler unit comprises a plurality of modules, each of the plurality of module divides an allocation number by two between two outputs of the module, and assigns any remainder between one of the two outputs that had not most recently been assigned a previous remainder.
5. The arbitration unit of Claim 1, wherein the scheduling unit comprises a plurality of modules, each including:
 - a divider;
 - a module memory that is capable of identifying an output which was previously assigned a first remainder; and

6 an adder that is capable of adding a second remainder to a number allocated to an output that had
7 previously not been assigned the first remainder.

1 6. The arbitration unit of Claim 5, wherein the divider divides a binary number by two
2 by stripping off the lower-order bit of the binary number.

1 7. The arbitration unit of Claim 5, wherein the module memory comprises:
2 a toggle flip-flop;
3 a plurality of gates that perform an ANDing function; and
4 an inverter.

8. The arbitration unit of Claim 1, wherein the binary distribution tree comprises P levels of modules, where 2^P is the maximum number of slots assignable to any frame.

9. The arbitration unit of Claim 8, wherein each module has two corresponding outputs.

1 10. The arbitration unit of Claim 9, wherein each module in the binary distribution tree
2 comprises:
3 a divider that divides an input number evenly between outputs of a module; and
4 a module memory that is capable of selecting which of the outputs of the module to transmit a
5 remainder.

1 11. The arbitration unit of Claim 10, wherein the module memory is configured to
2 alternate transmission of the remainder between the outputs of the module.

1 12. The arbitration unit of Claim 8, further comprising a vector processing unit that
2 converts a plurality of binary permission vectors into a single integer permission vectors.

1 13. The arbitration unit of Claim 12, wherein the vector processing unit comprises a
2 plurality of counter modules.

1 14. The arbitration unit of Claim 8, wherein a first level of scheduling module is capable
2 of processing an allocation number corresponding to a first flow while a second level of
3 scheduling module is processing an allocation number corresponding to a second flow.

1 15. A method for generating binary permission vectors, comprising:
2 setting a common drift direction at bifurcation points in a binary distribution tree;
3 dividing allocation numbers associated with a first flow by two at the bifurcation points;
4 assigning quotients from the dividing of the allocation numbers associated with the first
5 flow to branches at the bifurcation points;
6 assigning a remainder from the dividing of the allocation numbers associated with the
7 first flow to a branch that corresponds to a drift direction at the bifurcation point where the
8 remainder exists;
9 switching the drift direction at the bifurcation point where the remainder exists; and
10 reading outputs from the binary distribution tree as elements in a first binary permission
11 vector.

1 16. The method of Claim 15, wherein setting the common drift direction at the
2 bifurcation points in the binary distribution tree comprises assigning a leftward drift to scheduling
3 modules on a scheduling unit.

17. The method of Claim 15, wherein dividing the allocation numbers associated with the first flow comprises dividing an allocation number of the first flow and dividing values generated from dividing the allocation number of the first flow.

18. The method of Claim 15, further comprising:
dividing allocation numbers associated with a second flow by two at the bifurcation points;
assigning quotients from the dividing of the allocation numbers associated with the second flow to the branches at the bifurcation points;
assigning a remainder from the dividing of the allocation numbers associated with the second flow to a branch that corresponds to a drift direction at the bifurcation point where the remainder exists;
switching the drift direction at the bifurcation point where the remainder exists; and
reading the outputs from the binary distribution tree as elements in a second binary permission vector.

19. The method of Claim 18, wherein the dividing of allocation numbers associated with the second flow is performed concurrently with the dividing of allocation numbers associated with the first flow.

20. The method of Claim 18, further comprising generating an integer permission vector from the first binary permission vector and the second binary permission vector.

21. An arbitration unit, comprising:
means for assigning a first number of slots in a frame to a first flow and a second number of slots in the frame to a second flow; and

- 4 means for assigning first slot positions to the first number of slots in the frame and
- 5 second slot positions to the second number of slots in the frame using a binary distribution tree.

100232001
3232001
100232001